Federated Learning over Wireless Fading Channels

Mohammad Mohammadi Amiri, *Student Member, IEEE*, and Deniz Gündüz, *Senior Member, IEEE*

Abstract—We study federated machine learning at the wireless network edge, where limited power wireless devices, each with its own dataset, build a joint model with the help of a remote parameter server (PS). We consider a bandwidth-limited fading multiple access channel (MAC) from the wireless devices to the PS, and propose various techniques to implement distributed stochastic gradient descent (DSGD) over this shared noisy wireless channel. We first propose a digital DSGD (D-DSGD) scheme, in which one device is selected opportunistically for transmission at each iteration based on the channel conditions; the scheduled device quantizes its gradient estimate to a finite number of bits imposed by the channel condition, and transmits these bits to the PS in a reliable manner. Next, motivated by the additive nature of the wireless MAC, we propose a novel analog communication scheme, referred to as the *compressed analog* DSGD (CA-DSGD), where the devices first sparsify their gradient estimates while accumulating error from previous iterations, and project the resultant sparse vector into a low-dimensional vector for bandwidth reduction. We also design a power allocation scheme to align the received gradient vectors at the PS in an efficient manner. Numerical results show that D-DSGD outperforms other digital approaches in the literature; however, in general the proposed CA-DSGD algorithm converges faster than the D-DSGD scheme, and reaches a higher level of accuracy. We have observed that the gap between the analog and digital schemes increases when the datasets of devices are not independent and identically distributed (i.i.d.). Furthermore, the performance of the CA-DSGD scheme is shown to be robust against imperfect channel state information (CSI) at the devices. Overall these results show clear advantages for the proposed analog over-the-air DSGD scheme, which suggests that learning and communication algorithms should be designed jointly to achieve the best end-to-end performance in machine learning applications at the wireless edge.

I. Introduction

As the dataset sizes and model complexities grow, distributed machine learning (ML) is becoming the only viable alternative to centralized ML. In particular, with the increasing amount of information collected through wireless edge devices, such centralized solutions are becoming increasingly costly, due to the limited power and bandwidth available, and less desirable due to privacy concerns. Federated learning (FL) has been proposed as an alternative privacy-preserving distributed ML scheme, where each device participates in training using only locally available data with the help of

M. Mohammadi Amiri was with the Department of Electrical and Electronic Engineering, Imperial College London. He is now with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: mamiri@princeton.edu).

D. Gündüz is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: d.gunduz@imperial.ac.uk).

a parameter server (PS) [1]. In FL, devices exchange model parameters and their local updates with the PS, but the data never leaves the devices. In addition to privacy benefits, this is an attractive approach for wireless edge devices when dataset sizes are very large.

ML problems often involve the minimization of the empirical loss function

$$F\left(\boldsymbol{\theta}\right) = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{u} \in \mathcal{B}} f\left(\boldsymbol{\theta}, \boldsymbol{u}\right), \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ denotes the model parameters to be optimized, $\mathcal{B}$ is the training dataset of size $|\mathcal{B}|$ consisting of data samples and their labels, and $f\left(\cdot\right)$ is the loss function defined by the learning task. The minimization of $F\left(\boldsymbol{\theta}\right)$ is typically carried out through iterative stochastic gradient descent (SGD) algorithm, in which the model parameter vector at iteration t, $\boldsymbol{\theta}_t$, is updated with a stochastic gradient

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{g}\left(\boldsymbol{\theta}_t\right), \tag{2}$$

which satisfies $\mathbb{E}\left[\boldsymbol{g}\left(\boldsymbol{\theta}_t\right)\right] = \nabla F\left(\boldsymbol{\theta}_t\right)$, where η_t is the learning rate. SGD can easily be implemented across multiple devices, each of which has access to only a small fraction of the dataset. In distributed SGD (DSGD), at each iteration, device m computes a gradient vector based on the global parameter vector with respect to its local dataset, denoted by $\mathcal{B}_m$, and sends the result to the PS, which updates the global parameter vector according to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{g}_m\left(\boldsymbol{\theta}_t\right), \tag{3}$$

where M denotes the number of devices, and $\boldsymbol{g}_m\left(\boldsymbol{\theta}_t\right) \triangleq \frac{1}{|\mathcal{B}_m|} \sum_{\boldsymbol{u} \in \mathcal{B}_m} \nabla f\left(\boldsymbol{\theta}_t, \boldsymbol{u}\right), \; m \in [M]$. In FL, each device can carry out multiple local updates, and share the overall difference with respect to the previous global model with the PS [1].

What distinguishes FL from conventional ML is the large number of devices that participate in the training, and the low-capacity and unreliable links that connect these devices to the PS. Therefore, there have been significant research efforts to reduce the communication requirements in FL [1]–[24]. However, these and follow-up studies ignore the physical layer aspects of wireless connections and consider interference-and-error-free links from the participating devices to the PS, even though FL has been mainly motivated for mobile devices.

In this paper, we consider DSGD *over-the-air*; that is, we assume that learning takes place over a shared wireless medium over which the devices send their gradient estimates to the PS. To emphasize the limitations of the wireless medium,

we note that the dimension of some of the recent ML models, which also determines the size of the gradient estimates or model updates that must be transmitted to the PS at each iteration, can be extremely large, e.g., the 50-layer ResNet network has ~ 26 million weight parameters, while the VGGNet architecture has approximately 138 million parameters. On the other hand, available channel bandwidth is typically small due to the bandwidth and latency limitations; for example 1 LTE frame of 5MHz bandwidth and 10ms duration can carry only 6000 complex symbols. In principle, we can treat each iteration of the DSGD algorithm as a distributed over-the-air lossy computation problem. FL over a static Gaussian MAC is studied in [25], where both a digital scheme, which separates computation and communication, and an analog over-the-air computation scheme are introduced. While the digital scheme exploits gradient quantization followed by independent channel coding at the participating wireless devices, the analog scheme exploits the additive nature of the wireless channel and gradient sparsification, and employs random linear projection for dimensionality reduction. In [26] the authors consider a fading MAC, and also apply analog transmission, where each entry of a gradient vector at each of the devices is scheduled for transmission depending on the corresponding channel condition. A multi-antenna PS is considered in [27], where receive beamforming is used to maximize the number of devices scheduled for transmission at each iteration.

Here, we extend our previous works [25], [28], and study DSGD over a wireless fading MAC. While we focus on DSGD (also known as federated SGD), where each device sends its local gradient estimate at each iteration, the results can easily be extended by letting the devices send their model updates after several local SGD iterations. We first consider the *separate* computation and communication approach, and propose a digital DSGD (D-DSGD) scheme, in which only a single device is opportunistically scheduled for transmission at each iteration of DSGD based on the channel conditions from the devices to the PS. The scheduled device quantizes its gradient estimate to a finite number of bits using the gradient compression scheme in [18] while accumulating the error from previous iterations (this will be clarified later), and employs a channel code to transmit the bits over the available bandwidth-limited channel to the PS. For the MNIST classification task, it is shown that the proposed digital approach D-DSGD outperforms digital schemes that employ QSGD [8] or SignSGD [12] for gradient compression. We also observe that the proposed opportunistic scheduling scheme outperforms the scheme when all the devices participate in the transmission, with each device allocated orthogonal channel resources to communicate with the PS.

We then study analog transmission from the devices motivated by the signal-superposition property of the wireless MAC. At first, we extend the scheme in [26] by introducing error accumulation, which is shown to improve the performance. We then propose a novel scheme, inspired by the random projection used in [25] for dimensionality reduction, which we will refer to as the *compressed analog DSGD* (CA-DSGD). With CA-DSGD, we exploit the similarity in the sparsity patterns of the gradient estimates at different devices to speed up the computations, where each device projects its gradient estimate to a low-dimensional vector and transmits only the important gradient entries while accumulating the error. CA-DSGD provides the flexibility of adjusting the dimension of the gradient estimate sent by each device, which is particularly important for bandwidth-limited wireless channels, where the bandwidth may not be sufficient to send the entire gradient vector. A power allocation scheme is also designed, which aligns the vectors sent by different devices at the PS while satisfying the average power constraint. Numerical results show that the proposed CA-DSGD scheme improves upon the other analog and digital schemes under consideration with the same average power constraint and bandwidth resources, and the improvement is more significant when the datasets across devices are non-independent and identically distributed (i.i.d.). Its performance is also shown to be robust against imperfect channel state information (CSI) at the devices, whereas digital schemes are sensitive to accurate CSI, particularly if close to capacity operation is desired.

In addition to these benefits, we make the following observations:

1) The improvement of analog over-the-air computation compared to the D-DSGD scheme is particularly striking in the low power regime. This is mainly due to the "beamforming" effect of simultaneously transmitting highly correlated gradient estimates.

2) While both the convergence speed and the accuracy of the D-DSGD scheme increase significantly with the available average power, the performance of the analog schemes improve marginally. This highlights the energy efficiency of over-the-air computation, and makes it particularly attractive for FL across low-power IoT sensors.

3) Increasing the number of devices improves the accuracy for all the schemes even if the total dataset size and total power consumption remain the same. This "diversity gain" is much more limited for the analog scheme, and diminishes further as the training duration increases.

4) We observe that the performance of the CA-DSGD scheme improves if we reduce the bandwidth used at each iteration, and increase the number of iterations instead.

Notations: $\mathbb{R}$ and $\mathbb{C}$ represent the sets of real and complex values, respectively. For vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ with the same dimension, $\boldsymbol{x} \circ \boldsymbol{y}$ returns their Hadamard/entry-wise product. For a vector $\boldsymbol{z} \in \mathbb{C}^i$, $\mathrm{Re}\{\boldsymbol{z}\} \in \mathbb{R}^i$ and $\mathrm{Im}\{\boldsymbol{z}\} \in \mathbb{R}^i$ return the entry-wise real and imaginary components of $\boldsymbol{z}$, respectively. Also, $[\boldsymbol{v}, \boldsymbol{w}]$ represents the concatenation of two row vectors $\boldsymbol{v}$ and $\boldsymbol{w}$. We denote a zero-mean normal distribution with variance σ^2 by $\mathcal{N}\left(0, \sigma^2\right)$, and $\mathcal{CN}\left(0, \sigma^2\right)$ represents a complex normal distribution with independent real and imaginary terms each distributed according to $\mathcal{N}\left(0, \sigma^2/2\right)$. For positive integer i, we let $[i] \triangleq \{1, \ldots, i\}$. We denote the cardinality of set $\mathcal{A}$ by $|\mathcal{A}|$, and l_2 norm of vector $\boldsymbol{x}$ by $\|\boldsymbol{x}\|_2$. The imaginary unit is represented by j.

II. SYSTEM MODEL

We consider FL across M wireless devices, each with its own local dataset, which employ DSGD with the help of a
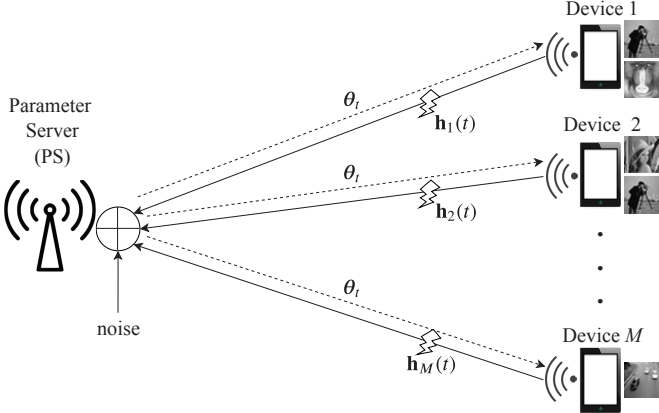
Fig. 1: Illustration of the wireless FL architecture.

remote PS. We model the channel from the devices to the PS as a wireless fading MAC, and OFDM is employed for transmission. The system model is illustrated in Fig. 1. The parameter vector at iteration t is denoted by $\boldsymbol{\theta}_t$, and we assume that it is delivered from the PS to the devices over an error-free shared link. We denote the set of data samples available at device m by $\mathcal{B}_m$, with $|\mathcal{B}_m| = B$, $\forall m \in [M]$, and the stochastic gradient computed by device m with respect to local data samples by $\boldsymbol{g}_m(\boldsymbol{\theta}_t) \in \mathbb{R}^d$, $m \in [M]$. At the t-th iteration of the DSGD algorithm in (3), the local gradient estimates of the devices are sent to the PS over a wireless fading MAC using s subchannels for a total of N time slots, where $s \leq d$ (in practice, we typically have $s \ll d$). We denote the length-s channel input vector transmitted by device m at the n-th time slot of the t-th iteration of the DSGD by $\boldsymbol{x}_m^n(t) = [x_{m,1}^n(t) \cdots x_{m,s}^n(t)]^T \in \mathbb{C}^s$. The channel output $\boldsymbol{y}^n(t) \in \mathbb{C}^s$ received by the PS at the n-th time slot of the t-th iteration, $n \in [N]$, is given by

$$\boldsymbol{y}^n(t) = \sum_{m=1}^M \boldsymbol{h}_m^n(t) \circ \boldsymbol{\mu}_m^n(t) \circ \boldsymbol{x}_m^n(t) + \boldsymbol{z}^n(t), \qquad (4)$$

where $\boldsymbol{\mu}_m^n(t) \in \{0,1\}^s$ is the entry-wise scheduling vector with the i-th entry $\mu_{m,i}^n(t) = 1$, if $m \in \mathcal{M}_i^n(t)$, and $\mu_{m,i}^n(t) = 0$, otherwise[1], $\boldsymbol{h}_m^n(t) \in \mathbb{C}^s$ is the channel gains vector from device m to the PS with the i-th entry $h_{m,i}^n(t)$ i.i.d. according to $\mathcal{CN}(0, \sigma^2)$, e.g., Rayleigh fading, and $\boldsymbol{z}^n(t) \in \mathbb{C}^s$ is complex Gaussian noise vector with the i-th entry $z_i^n(t)$ i.i.d. according to $\mathcal{CN}(0,1)$. The channel input vector of device m at the n-th time slot of iteration t, $n \in [N]$, is a function of the channel gain vector $\boldsymbol{h}_m^n(t)$, current parameter vector $\boldsymbol{\theta}_t$, the local dataset $\mathcal{B}_m$, and the current gradient estimate at device m, $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$, $m \in [M]$. We assume that, at each time slot, the CSI is known by the devices and the PS. For a total of T iterations of the DSGD algorithm, the following total average transmit power constraint is imposed at device m:

$$\frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \mathbb{E}\left[\|\boldsymbol{x}_m^n(t)\|_2^2 \right] \leq \bar{P}, \quad \forall m \in [M], \qquad (5)$$

[1]$\mathcal{M}_i^n(t)$ is a subset of the devices that will be specified for each of the schemes.

where the expectation is taken over the randomness of the channel gains.

The goal is to recover $\frac{1}{M}\sum_{m=1}^M \boldsymbol{g}_m(\boldsymbol{\theta}_t)$ at the PS, which then updates the model parameter as in (3) after N time slots. However, due to the pre-processing performed at each device and the distortion caused by the wireless channel, the PS uses a noisy estimate to update the model parameter. Having defined $\boldsymbol{y}(t) \triangleq [\boldsymbol{y}^1(t)^T \cdots \boldsymbol{y}^N(t)^T]^T$, we have $\boldsymbol{\theta}_{t+1} = \phi(\boldsymbol{\theta}_t, \boldsymbol{y}(t))$ for some update function $\phi : \mathbb{R}^d \times \mathbb{C}^{Ns} \to \mathbb{R}^d$. The updated model parameter is then multicast to the devices by the PS through an error-free shared link, so the devices receive a consistent parameter vector for their computations in the next iteration.

Note that the goal is to recover the average of the local gradient estimates at the PS, which is a distributed lossy computation problem over a noisy MAC. We will consider both a digital approach based on separating computations and communication, and an analog approach, where the gradients are transmitted in an analog fashion, without being converted into bits first. Analog transmission has been well studied for image/ video multicasting over wireless channels in recent years [29]–[31], and here we employ the random projection technique proposed in [31] for image transmission over a bandwidth limited wireless channel.

III. DIGITAL DSGD

We first consider DSGD with digital transmission of the gradient estimates by the devices over the wireless fading MAC, referred to as the digital DSGD (D-DSGS) scheme. For D-DSGD, we consider $N = 1$, i.e., the parameter vector is updated after each time slot, and drop the dependency on time slot parameter n.

The goal here is to schedule devices and employ power allocation across time slots such that devices can transmit to the PS their local gradient estimates as accurately as possible. A possible approach is to schedule all the devices at all the iterations; however, due to the interference among the devices this will result in each device sending a very coarse description of its local gradient estimate. Instead here we will schedule the devices opportunistically according to their channel states.

In particular, with the knowledge of channel state information (CSI), at each iteration t, we select the device with the largest value of $\sum_{i=1}^s |h_{m,i}(t)|^2$, $m \in [M]$. Accordingly, the index of the transmitting device at iteration t is given by:

$$m^*(t) = \arg \max_{m \in [M]} \left\{ \sum_{i=1}^s |h_{m,i}(t)|^2 \right\}. \qquad (6)$$

We note that, due to the symmetry in the model, the probability of selecting a device at any time is the same, $1/M$. The power allocated to device m at the t-th iteration is given by $\bar{P}_m(t)$, where $\bar{P}_m(t) = 0$, if $m \neq m^*(t)$, and it should satisfy

$$\frac{1}{MT} \sum_{t=1}^T \bar{P}_m(t) \leq \bar{P}, \quad \text{for } m \in [M]. \qquad (7)$$

For the rate of transmission, we will use a capacity upper bound. The i-th entry of the channel output at the t-th iteration is given by

$$y_i(t) = h_{m^*(t),i}(t) \, x_{m^*(t),i}(t) + z_i(t), \quad i \in [s], \qquad (8)$$

which is equivalent to a wireless fast fading channel with a limited number of s channel uses, with CSI known at both the transmitter and receiver. In the following, we provide an upper bound on the capacity of this channel by treating it as s parallel Gaussian channels. This is equivalent to coding across infinitely many realizations of this s-dimensional channel. For a transmit power $P_{m^*(t)}(t)$, the capacity of this parallel Gaussian channel is obtained as the solution of the following optimization problem [32, Section 5.4.6]:

$$\max_{P_1,\ldots,P_s} \sum_{i=1}^{s} \log_2\left(1 + P_i\left|h_{m^*(t),i}(t)\right|^2\right),$$
$$\text{subject to } \sum_{i=1}^{s} P_i = \bar{P}_{m^*(t)}(t). \tag{9}$$

The optimization problem in (9) is solved through waterfilling, and the optimal power allocation is given by

$$P_i^* = \max\left\{\frac{1}{\zeta} - \frac{1}{\left|h_{m^*(t),i}(t)\right|^2}, 0\right\}, \tag{10}$$

where ζ is determined such that $\sum_{i=1}^{s} P_i^* = \bar{P}_{m^*(t)}(t)$. Having calculated $P_1^*,\ldots,P_s^*$, the capacity of the wireless channel in (8) is given by

$$R(t) = \sum_{i=1}^{s} \log_2\left(1 + P_i^*\left|h_{m^*(t),i}(t)\right|^2\right), \tag{11}$$

which provides an upper bound on the capacity of the communication channel between device $m^*(t)$ and the PS. We would like to emphasize that this capacity upper bound can be quite loose especially for small s values.

We adopt the D-DSGD scheme proposed in [25, Section III], in which the gradient estimate $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$, computed at device m, is added to the error accumulated from previous iterations, denoted by $\boldsymbol{\Delta}_m(t-1)$, where we set $\boldsymbol{\Delta}_m(0) = \boldsymbol{0}$, $m \in [M]$. For the compression of the error compensated gradient vector $\boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1)$, we employ the scheme in [18], where it is first sparsified by setting all but the highest $q(t)$ positive and the smallest $q(t)$ negative entries to zero, where $q(t) \leq d/2$ (in practice, the goal is to have $q(t) \ll d$, $\forall t$). Then, device m computes the mean value of the positive and negative entries of the resultant sparse vector, denoted by $q_m^+(t)$ and $q_m^-(t)$, respectively, $m \in [M]$. If $q_m^+(t) \geq q_m^-(t)$, device m sets all the negative entries of the sparse vector to zero and all the positive entries to $q_m^+(t)$, and vice versa, if $q_m^+(t) < q_m^-(t)$, $m \in [M]$. Let $\hat{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$ denote the resultant sparse vector at device m, $m \in [M]$. After computing $\hat{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$ at device m, $m \in [M]$, the error accumulation vector, which maintains those entries of vector $\boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1)$ that are not transmitted, is updated as follows:

$$\boldsymbol{\Delta}_m(t) = \begin{cases} \boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1) - \hat{\boldsymbol{g}}_m(\boldsymbol{\theta}_t), & \text{if } m = m^*(t), \\ \boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1), & \text{otherwise.} \end{cases}$$

We note that, if user m is scheduled, the accumulated error at device m is the difference between $\boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1)$ and its sparsified version $\hat{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$, $m \in [M]$; on the other hand, if device m is not scheduled, we maintain vector $\boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m(t-1)$ entirely as the accumulated error. For a sparsity level $q(t)$, the D-DSGD scheme requires transmission of a total of [25, Equation (10)]

$$r(t) = \log_2 \binom{d}{q(t)} + 33 \text{ bits.} \tag{12}$$

We assume that device $m^*(t)$ employs a capacity achieving channel code using the optimal value of the capacity upper bound in (11), and we set the sparsity level $q(t)$ as the highest integer satisfying $r(t) \leq R(t)$.

We highlight here that with the proposed D-DSGD algorithm, only a single device is scheduled for transmission according to (6). The PS updates the parameter vector after receiving the gradient estimate from the scheduled device and shares it with all the devices to continue their computations.

Remark 1. *Alternatively, we can select the device with the highest capacity upperbound; however this would introduce an overhead as the PS will need to solve the waterfilling power allocation for all M devices. This will be prohibitive for large s and M values.*

Remark 2. *We can also schedule all or a subset of the devices, and allocate distinct subchannels to scheduled devices. In Section V we consider the so-called orthogonal digital DSGD (OD-DSGD) scheme, which schedules all the devices at each iteration, where each device is allocated $\lfloor s/M \rfloor$ distinct subchannels. We have observed that OD-DSGD performs much worse than D-DSGD. It is worth noting that scheduling multiple devices reduces the number of subchannels allocated to each device, and forces the devices to transmit their information at shorter blocklengths. In practice, this would result in a higher error probability or reduced transmission rate [33]. An alternative approach is to code across time slots by allocating multiple time slots to a scheduled user. This requires the information about the future channel gains, which is not possible in our model since the channel gains are assumed i.i.d. across time slots and users.*

We will evaluate the performance of D-DSGD in Section V, and study in detail the impact of various system parameters, such as the power constraint and the number of devices on the performance. We will also compare D-DSGD with other compression schemes in the literature, as well as the analog transmission of local gradients, which we present next.

IV. ANALOG DSGD

Analog DSGD is motivated by the fact that the PS is only interested in the average of the gradient vectors, and the underlying wireless MAC can provide the sum of the gradients if they are sent in an uncoded fashion. We first present a generalization of the over-the-air computation approach introduced in [26], referred to as *entry-wise scheduled analog DSGD (ESA-DSGD)*, and then extend it by introducing error accumulation, referred to as *error compensated ESA-DSGD (ECESA-DSGD)*. Finally, we propose a novel analog scheme, built upon our previous work [25], referred to as *compressed analog DSGD (CA-DSGD)*.

A. ESA-DSGD

With the ESA-DSGD scheme studied in [26], each device sends its gradient estimate entirely after applying power allo-

cation, which is to satisfy the average power constraint. At the t-th iteration of the DSGD, device m, $m \in [M]$, transmits its local gradient estimate $\boldsymbol{g}_m(\boldsymbol{\theta}_t) \in \mathbb{R}^d$ over $N = \lceil d/2s \rceil$ time slots by utilizing both the real and imaginary components of the available s subchannels. We define, for $n \in [N]$, $m \in [M]$,

$$\boldsymbol{g}_{m,\text{re}}^n(\boldsymbol{\theta}_t) \triangleq [g_{m,2(n-1)s+1}(\boldsymbol{\theta}_t), \cdots, g_{m,(2n-1)s}(\boldsymbol{\theta}_t)]^T, \tag{13a}$$

$$\boldsymbol{g}_{m,\text{im}}^n(\boldsymbol{\theta}_t) \triangleq [g_{m,(2n-1)s+1}(\boldsymbol{\theta}_t), \cdots, g_{m,2ns}(\boldsymbol{\theta}_t)]^T, \tag{13b}$$

$$\boldsymbol{g}_m^n(\boldsymbol{\theta}_t) \triangleq \boldsymbol{g}_{m,\text{re}}^n(\boldsymbol{\theta}_t) + j\boldsymbol{g}_{m,\text{im}}^n(\boldsymbol{\theta}_t), \tag{13c}$$

where $g_{m,i}(\boldsymbol{\theta}_t)$ is the i-th entry of $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$, and we zero-pad $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$ to dimension $2sN$. We note that, according to (13),

$$\boldsymbol{g}_m(\boldsymbol{\theta}_t) = [\boldsymbol{g}_{m,\text{re}}^1(\boldsymbol{\theta}_t), \boldsymbol{g}_{m,\text{im}}^1(\boldsymbol{\theta}_t), \cdots, \boldsymbol{g}_{m,\text{re}}^N(\boldsymbol{\theta}_t), \boldsymbol{g}_{m,\text{im}}^N(\boldsymbol{\theta}_t)]^T, \tag{14}$$

where $N = \lceil d/2s \rceil$. At the n-th time slot of the t-th iteration, device m, $m \in [M]$, sends $\boldsymbol{x}_m^n(t) = \boldsymbol{\alpha}_m^{\text{e},n}(t) \circ \boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{\text{e},n}(t) \in \mathbb{C}^s$ is the power allocation vector, which is set to satisfy the average transmit power constraint. Thus, after N time slots, each device sends its gradient estimate of dimension d entirely. The i-th entry of the power allocation vector $\boldsymbol{\alpha}_m^{\text{e},n}(t)$ is set as follows:

$$\alpha_{m,i}^{\text{e},n}(t) = \begin{cases} \frac{\gamma_m^{\text{e},n}(t)}{h_{m,i}^n(t)}, & \text{if } |h_{m,i}^n(t)|^2 \geq \lambda^{\text{e}}(t), \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

for some $\gamma_m^{\text{e},n}(t), \lambda^{\text{e}}(t) \in \mathbb{R}$, set to satisfy the average transmit power constraint. According to (15), each entry of a gradient vector is transmitted if its corresponding channel gain is over a threshold. The set of devices selected to transmit the i-th entry of the channel input vector at the n-th time slot is given by, $i \in [s]$, $n \in [N]$,

$$\mathcal{M}_i^n(t) = \left\{ m \in [M] : |h_{m,i}^n(t)|^2 \geq \lambda^{\text{e}}(t) \right\}. \tag{16}$$

In the following, we analyze the average transmit power of the ESA-DSGD scheme based on the power allocation design given in (15). We set the parameters $\gamma_m^{\text{e},n}(t)$ and $\lambda^{\text{e}}(t)$ to obtain the same average transmit power $\bar{P}^n(t)$ at device m, $m \in [M]$, in time slot n, $n \in [N]$, of iteration t, which satisfies

$$\frac{1}{NT} \sum_{t=1}^T \sum_{n=1}^N \bar{P}^n(t) \leq \bar{P}. \tag{17}$$

According to (15), $\forall m \in [M]$, we have, for $n \in [N], t \in [T]$,

$$\bar{P}^n(t) = \mathbb{E}\left[\|\boldsymbol{x}_m^n(t)\|_2^2\right]$$

$$= \sum_{i=1}^s \mathbb{E}\left[|\alpha_{m,i}^{\text{e},n}(t)|^2 |g_{m,(n-1)s+i}(\boldsymbol{\theta}_t)|^2\right]. \tag{18}$$

We highlight that the entries of the gradient vector $\boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$ are independent of the channel gains $h_{m,i}^n(t)$, $\forall i, n, m$. Since the power allocation vector $\boldsymbol{\alpha}_m^{\text{e},n}(t)$ is a function of $\boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$, it follows that, for $n \in [\lceil d/2s \rceil]$, $m \in [M]$,

$$\bar{P}^n(t) = \sum_{i=1}^s |g_{m,(n-1)s+i}(\boldsymbol{\theta}_t)|^2 \mathbb{E}\left[|\alpha_{m,i}^{\text{e},n}(t)|^2\right]. \tag{19}$$

Note that $|h_{m,i}^n(t)|^2$ follows an exponential distribution with mean σ^2, $\forall i, n, m$. Thus, we have

$$\mathbb{E}\left[|\alpha_{m,i}^{\text{e},n}(t)|^2\right] = \left(\frac{\gamma_m^{\text{e},n}(t)}{\sigma}\right)^2 \text{E}_1(\lambda^{\text{e}}(t)), \tag{20}$$

where $\text{E}_1(x) \triangleq \int_x^\infty \frac{e^{-\tau}}{\tau} d\tau$. It follows that, $m \in [M], n \in [N]$,

$$\bar{P}^n(t) = \left(\frac{\gamma_m^{\text{e},n}(t)}{\sigma}\right)^2 \text{E}_1(\lambda^{\text{e}}(t)) P_m^{\text{e},n}(t), \tag{21}$$

where we define $P_m^{\text{e},n}(t) \triangleq \|\boldsymbol{g}_m^n(\boldsymbol{\theta}_t)\|_2^2$. Given the threshold value $\lambda^{\text{e}}(t)$, we set, for $m \in [M], n \in [N], t \in [T]$,

$$\gamma_m^{\text{e},n}(t) = \sigma\left(\frac{\bar{P}^n(t)}{\text{E}_1(\lambda^{\text{e}}(t)) P_m^{\text{e},n}(t)}\right)^{1/2}, \tag{22}$$

which we note that it does not differ significantly across devices, since values of $P_m^{\text{e},n}(t)$, $\forall m$, are not too different. We assume that, before transmitting $\boldsymbol{x}_m^n(t) = \boldsymbol{\alpha}_m^{\text{e},n}(t) \circ \boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$, device m, $m \in [M]$, sends $\gamma_m^{\text{e},n}(t)$ to the PS in an error-free fashion using an error correcting code, and the PS computes

$$\bar{\gamma}^{\text{e},n}(t) \triangleq \frac{1}{M} \sum_{m=1}^M \gamma_m^{\text{e},n}(t), \quad n \in [N], t \in [T]. \tag{23}$$

This factor is used by the PS to scale down the received signal.

Here we analyze the received signal at the PS. By substituting $\boldsymbol{x}_m^n(\boldsymbol{\theta}_t)$ and $\boldsymbol{\alpha}_m^{\text{e},n}(t)$ into (4), it follows that, for $i \in [s]$, $n \in [N]$,

$$y_i^n(t) = \sum_{m \in \mathcal{M}_i^n(t)} \gamma_m^{\text{e},n}(t) \left(g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)\right.$$

$$\left. + jg_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)\right) + z_i^n(t). \tag{24}$$

The PS has perfect CSI, and hence, knows set $\mathcal{M}_i^n(t)$. Its goal is to recover $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m \in \mathcal{M}_i^n(t)} g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$ and $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m \in \mathcal{M}_i^n(t)} g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$, which provide estimates for $\frac{1}{M} \sum_{m=1}^M g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$ and $\frac{1}{M} \sum_{m=1}^M g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$, respectively. The PS estimates $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m \in \mathcal{M}_i^n(t)} g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t)$, for $i \in [s]$, $n \in [N]$, using its noisy observation $y_i^n(t)$, given in (24), as

$$\hat{g}_{2(n-1)s+i}^{\text{e}}(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Re}\{y_i^n(t)\}}{\bar{\gamma}^{\text{e},n}(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

and estimates $\frac{1}{|\mathcal{M}_i^n(t)|} \sum_{m \in \mathcal{M}_i^n(t)} g_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)$ through

$$\hat{g}_{(2n-1)s+i}^{\text{e}}(\boldsymbol{\theta}_t) = \begin{cases} \frac{\text{Im}\{y_i^n(t)\}}{\bar{\gamma}^{\text{e},n}|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise}. \end{cases} \tag{26}$$

Estimated vector $\hat{\boldsymbol{g}}^{\text{e}}(\boldsymbol{\theta}_t) \triangleq [\hat{g}_1^{\text{e}}(\boldsymbol{\theta}_t) \cdots \hat{g}_d^{\text{e}}(\boldsymbol{\theta}_t)]^T$ is then used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\boldsymbol{g}}^{\text{e}}(\boldsymbol{\theta}_t)$.

Remark 3. *We remark here that the scheme in [26] imposes a stricter average power constraint $\bar{P}$ per iteration of the DSGD, i.e., at device m we should have*

$$\frac{1}{N} \sum_{n=1}^N \mathbb{E}\left[\|\boldsymbol{x}_m^n(t)\|_2^2\right] \leq \bar{P}, \quad \forall m \in [M], \forall t. \tag{27}$$

For fairness in our comparisons we relax this power constraint, and impose the one in (5)*, which constrains the average power over all the iterations.*

B. ECESA-DSGD

With the ESA-DSGD scheme, entries of the gradient vectors that are not sent due to poor channel conditions are completely forgotten. The proposed ECESA-DSGD scheme modifies ESA-DSGD by incorporating error accumulation technique to retain the accuracy of local gradients.

We denote the error accumulation vector of device m at the n-th time slot of the t-iteration by $\boldsymbol{\Delta}_m^{v,n}(t) \in \mathbb{C}^s$, and set $\boldsymbol{\Delta}_m^{v,n}(t) = \mathbf{0}$, $\forall n, t, m$. Similarly to the ESA-DSGD scheme, with ECESA-DSGD, each device sends its entire gradient estimate of dimension d through $N = \lceil d/2s \rceil$ time slots, where the gradient estimates at the devices are zero-padded to dimension $2sN$. After computing $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$ and obtaining $\boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$ according to (13), device m, $m \in [M]$, updates its gradient estimate with the accumulated error as $\boldsymbol{g}_m^{v,n}(\boldsymbol{\theta}_t) \triangleq \boldsymbol{g}_m^n(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m^{v,n}(t-1)$, for $n \in [N]$, and transmits vector $\boldsymbol{x}_m^n(t) = \boldsymbol{\alpha}_m^{v,n}(t) \circ \boldsymbol{g}_m^{v,n}(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{v,n}(t) \in \mathbb{C}^s$ is the power allocation vector, whose i-th entry is given by:

$$\alpha_{m,i}^{v,n}(t) = \begin{cases} \frac{\gamma_m^{v,n}(t)}{h_{m,i}^n(t)}, & \text{if } \left|h_{m,i}^n(t)\right|^2 \geq \lambda^v(t), \\ 0, & \text{otherwise,} \end{cases} \quad (28)$$

for some $\gamma_m^{v,n}(t), \lambda^v(t) \in \mathbb{R}$. Device m, $m \in [M]$, then updates the i-th entry of vector $\boldsymbol{\Delta}_m^{v,n}(t)$ as follows:

$$\begin{aligned} \Delta_{m,i}^{v,n}(t) &= \left(1 - \mathbb{1}\left(\alpha_{m,i}^{v,n}(t) \neq 0\right)\right) g_{m,i}^n(\boldsymbol{\theta}_t) \\ &= \left(1 - \mathbb{1}\left(\alpha_{m,i}^{v,n}(t) \neq 0\right)\right) \left(g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) \right. \\ &\qquad\qquad \left. + jg_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)\right), \quad (29) \end{aligned}$$

where $\mathbb{1}(\cdot)$ is the indicator function, and $g_{m,i}^n(\boldsymbol{\theta}_t)$ denotes the i-th entry of $\boldsymbol{g}_m^n(\boldsymbol{\theta}_t)$, for $i \in [s]$, $n \in [N]$. Thus, the i-th entry of vector $\boldsymbol{g}_m^{v,n}(\boldsymbol{\theta}_t)$ is given by, for $i \in [s]$, $n \in [N]$,

$$\begin{aligned} g_{m,i}^{v,n}(\boldsymbol{\theta}_t) &= g_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) + jg_{m,(2n-1)s+i}(\boldsymbol{\theta}_t) \\ &\quad + \left(1 - \mathbb{1}\left(\alpha_{m,i}^{v,n}(t-1) \neq 0\right)\right)\left(g_{m,2(n-1)s+i}(\boldsymbol{\theta}_{t-1})\right. \\ &\qquad\qquad \left. + jg_{m,(2n-1)s+i}(\boldsymbol{\theta}_{t-1})\right). \quad (30) \end{aligned}$$

According to (29), each entry of the gradient vector $\boldsymbol{g}_m^{v,n}(\boldsymbol{\theta}_t)$ that is not transmitted due to the power allocation given in (28), is retained in the error accumulation vector $\boldsymbol{\Delta}_m^{v,n}(t)$ for possible transmission in the next iteration.

Here we provide the power analysis of the ECESA-DSGD scheme. For fairness, we set the parameters $\gamma_m^{v,n}(t)$ and $\lambda^v(t)$ yielding an average transmit power $\bar{P}^n(t)$ at device m, $m \in [M]$, in time slot n, $n \in [N]$, of iteration t, satisfying the constraint in (17). Since the power allocation vector of ECESA-DSGD, given in (28), is similar to that of the ESA-DSGD, by following a similar procedure we obtain the following average power at device m for ECESA-DSGD:

$$\bar{P}^n(t) = \left(\frac{\gamma_m^{v,n}(t)}{\sigma}\right)^2 \mathrm{E}_1(\lambda^v(t))P_m^{v,n}(t), \quad n \in [N], \quad (31)$$

where we define $P_m^{v,n}(t) \triangleq \|\boldsymbol{g}_m^{v,n}(\boldsymbol{\theta}_t)\|_2^2$, $t \in [T]$. For a fixed $\lambda^v(t)$, we set, for $m \in [M], n \in [N], t \in [T]$,

$$\gamma_m^{v,n}(t) = \sigma \left(\frac{\bar{P}^n(t)}{\mathrm{E}_1(\lambda^e(t))P_m^{v,n}(t)}\right)^{1/2}, \quad (32)$$

shared with the PS in an error-free manner, through which the PS computes

$$\bar{\gamma}^{v,n}(t) \triangleq \frac{1}{M}\sum_{m=1}^M \gamma_m^{v,n}(t), \quad n \in [N], t \in [T]. \quad (33)$$

From the power allocation in (28), it follows that, for $i \in [s]$, $n \in [N]$,

$$y_i^n(t) = \sum_{m \in \mathcal{M}_i^n(t)} \gamma_m^{v,n}(t)g_{m,i}^{v,n}(\boldsymbol{\theta}_t) + z_i^n(t), \quad (34)$$

where we have

$$\mathcal{M}_i^n(t) = \left\{m \in [M] : \left|h_{m,i}^n(t)\right|^2 \geq \lambda^v(t)\right\}. \quad (35)$$

Having perfect CSI, the PS's goal is to recover $\frac{1}{|\mathcal{M}_i^n(t)|}\sum_{m \in \mathcal{M}_i^n(t)} g_{m,i}^{v,n}(\boldsymbol{\theta}_t)$, the real and imaginary terms of which provide estimates for $\frac{1}{M}\sum_{m=1}^M g_{m,2(n-1)s+i}^n(\boldsymbol{\theta}_t)$ and $\frac{1}{M}\sum_{m=1}^M g_{m,(2n-1)s+i}^n(\boldsymbol{\theta}_t)$, respectively, for $i \in [s]$, $n \in [N]$. The PS estimates $\frac{1}{M}\sum_{m=1}^M g_{m,2(n-1)s+i}^n(\boldsymbol{\theta}_t)$ as

$$\hat{g}_{2(n-1)s+i}^v(\boldsymbol{\theta}_t) = \begin{cases} \frac{\mathrm{Re}\{y_i^n(t)\}}{\bar{\gamma}^{v,n}(t)\left|\mathcal{M}_i^n(t)\right|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ \hat{g}_{2(n-1)s+i}^v(\boldsymbol{\theta}_{t-1}), & \text{otherwise,} \end{cases}$$

$$(36)$$

and estimates $\frac{1}{M}\sum_{m=1}^M g_{m,(2n-1)s+i}^n(\boldsymbol{\theta}_t)$ through

$$\hat{g}_{(2n-1)s+i}^v(\boldsymbol{\theta}_t) = \begin{cases} \frac{\mathrm{Im}\{y_i^n(t)\}}{\bar{\gamma}^{v,n}(t)\left|\mathcal{M}_i^n(t)\right|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ \hat{g}_{(2n-1)s+i}^v(\boldsymbol{\theta}_{t-1}), & \text{otherwise,} \end{cases}$$

$$(37)$$

for $i \in [s]$, $n \in [N]$. After $N = \lceil d/2s \rceil$ time slots, estimated vector $\hat{\boldsymbol{g}}^v(\boldsymbol{\theta}_t) \triangleq [\hat{g}_1^v, \cdots, \hat{g}_d^v]^T$ is then used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t\hat{\boldsymbol{g}}^v(\boldsymbol{\theta}_t)$.

C. CA-DSGD

As opposed to ESA-DSGD and ECESA-DSGD, which aim to transmit all the gradient entries to the PS at each DSGD iteration, i.e., $N = \lceil d/2s \rceil$, the CA-DSGD scheme in Algorithm 1 reduces the transmission bandwidth by a linear projection. Each device projects its gradient estimate to dimension $\tilde{s} = 2sN$, which can then be transmitted through N time slots, for some $N \in [\lceil d/2s \rceil]$.

We describe the CA-DSGD scheme for an arbitrary number of time slots $N \in [\lceil d/2s \rceil]$ per iteration of DSGD, which is determined later. At each iteration the devices sparsify their gradient estimates as described below. They employ *error accumulation* [7], where the accumulated error vector at device m until iteration t is denoted by $\boldsymbol{\Delta}_m^c(t-1) \in \mathbb{R}^d$, where we set $\boldsymbol{\Delta}_m^c(0) = \mathbf{0}$, $\forall m \in [M]$. After computing $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$, device m updates its estimate with the accumulated error as $\boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t) \triangleq \boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m^c(t-1)$, $m \in [M]$. Next, the devices apply gradient sparsification, where device m sets

Algorithm 1 CA-DSGD

1: **Initialize** $\boldsymbol{\theta}_1 = \mathbf{0}$ and $\boldsymbol{\Delta}_1^{\mathrm{c}}(0) = \cdots = \boldsymbol{\Delta}_M^{\mathrm{c}}(0) = \mathbf{0}$
2: **for** $t = 1, \ldots, T$ **do**
 • **devices do:**
3:    **for** $m = 1, \ldots, M$ in parallel **do**
4:      Compute $\boldsymbol{g}_m(\boldsymbol{\theta}_t)$ with respect to local dataset $\mathcal{B}_m$
5:      $\boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t) = \boldsymbol{g}_m(\boldsymbol{\theta}_t) + \boldsymbol{\Delta}_m^{\mathrm{c}}(t-1)$
6:      $\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t) = \mathrm{sparse}_k(\boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t))$
7:      $\boldsymbol{\Delta}_m^{\mathrm{c}}(t) = \boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t) - \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$
8:      $\tilde{\boldsymbol{g}}_m(\boldsymbol{\theta}_t) = \boldsymbol{A}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$
9:      **for** $n = 1, \ldots, N$ **do**
10:        $\boldsymbol{x}_m^n(t) = \boldsymbol{\alpha}_m^{\mathrm{c},n}(t) \circ \tilde{\boldsymbol{g}}_m^n(\boldsymbol{\theta}_t)$
11:      **end for**
12:    **end for**
 • **PS does:**
13:    **if** $\hat{\boldsymbol{y}}(t) \neq \mathbf{0}$ **then**
14:      $\hat{\boldsymbol{g}}^{\mathrm{c}}(\boldsymbol{\theta}_t) = \mathrm{AMP}_{\boldsymbol{A}}(\hat{\boldsymbol{y}}(t))$
15:      $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\boldsymbol{g}}^{\mathrm{c}}(\boldsymbol{\theta}_t)$
16:    **else**
17:      $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$
18:    **end if**
19: **end for**

all but k elements with the highest magnitudes of $\boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t)$ to zero, where $k \leq \tilde{s}$ is a design parameter, and obtains a sparse vector $\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$, $m \in [M]$. This k-level sparsification is represented by function sparse_k in Algorithm 1, i.e., $\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t) = \mathrm{sparse}_k(\boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t))$. Device m, $m \in [M]$, then updates $\boldsymbol{\Delta}_m^{\mathrm{c}}(t)$ as $\boldsymbol{\Delta}_m^{\mathrm{c}}(t) = \boldsymbol{g}_m^{ec}(\boldsymbol{\theta}_t) - \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$. To transmit the sparse vectors over the limited-bandwidth channel, devices employ a random projection matrix.

A pseudo-random matrix $\boldsymbol{A} \in \mathbb{R}^{\tilde{s} \times d}$, with each entry i.i.d. according to $\mathcal{N}(0, 1/\tilde{s})$, is generated and shared between the PS and the devices, where $\tilde{s} = 2sN$, for an arbitrary $N \in [\lceil d/2s \rceil]$. At each iteration t, device m computes $\tilde{\boldsymbol{g}}_m(\boldsymbol{\theta}_t) \triangleq \boldsymbol{A}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t) \in \mathbb{R}^{\tilde{s}}$, and aims to transmit it to the PS over $N = \tilde{s}/2s$ time slots. We define, for $n \in [N]$, $m \in [M]$,

$$\tilde{\boldsymbol{g}}_{m,\mathrm{re}}^n(\boldsymbol{\theta}_t) \triangleq [\tilde{g}_{m,2(n-1)s+1}(\boldsymbol{\theta}_t), \cdots, \tilde{g}_{m,(2n-1)s}(\boldsymbol{\theta}_t)]^T, \tag{38a}$$

$$\tilde{\boldsymbol{g}}_{m,\mathrm{im}}^n(\boldsymbol{\theta}_t) \triangleq [\tilde{g}_{m,(2n-1)s+1}(\boldsymbol{\theta}_t), \cdots, \tilde{g}_{m,2ns}(\boldsymbol{\theta}_t)]^T, \tag{38b}$$

$$\tilde{\boldsymbol{g}}_m^n(\boldsymbol{\theta}_t) \triangleq \tilde{\boldsymbol{g}}_{m,\mathrm{re}}^n(\boldsymbol{\theta}_t) + j\tilde{\boldsymbol{g}}_{m,\mathrm{im}}^n(\boldsymbol{\theta}_t), \tag{38c}$$

where $\tilde{g}_{m,i}(\boldsymbol{\theta}_t)$ is the i-th entry of $\tilde{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$, $i \in [\tilde{s}]$. At the n-th time slot of the t-th iteration of DSGD, device m, $m \in [M]$, sends $\boldsymbol{x}_m^n(t) = \boldsymbol{\alpha}_m^{\mathrm{c},n}(t) \circ \tilde{\boldsymbol{g}}_m^n(\boldsymbol{\theta}_t)$, where $\boldsymbol{\alpha}_m^{\mathrm{c},n}(t) \in \mathbb{C}^s$ is the power allocation vector. The i-th entry of the power allocation vector $\boldsymbol{\alpha}_m^{\mathrm{c},n}(t)$ is set as follows:

$$\alpha_{m,i}^{\mathrm{c},n}(t) = \begin{cases} \frac{\gamma_m^{\mathrm{c},n}(t)}{h_{m,i}^n(t)}, & \text{if } |h_{m,i}^n(t)|^2 \geq \lambda^{\mathrm{c}}(t), \\ 0, & \text{otherwise}, \end{cases} \tag{39}$$

for some $\gamma_m^{\mathrm{c},n}(t), \lambda^{\mathrm{c}}(t) \in \mathbb{R}$. The set of devices scheduled to transmit the i-th entry of the channel input vector at the n-th time slot is given by, $i \in [s]$, $n \in [N]$,

$$\mathcal{M}_i^n(t) = \left\{ m \in [M] : |h_{m,i}^n(t)|^2 \geq \lambda_m^{\mathrm{c},n}(t) \right\}. \tag{40}$$

Similarly to ESA-DSGD and ECESA-DSGD, we set the average transmit power at device m, $m \in [M]$, in time slot n, $n \in [N]$, of iteration t for CA-DSGD to $\bar{P}^n(t)$,

$$\bar{P}^n(t) = \left(\frac{\gamma_m^{\mathrm{c},n}(t)}{\sigma}\right)^2 \mathrm{E}_1(\lambda^{\mathrm{c}}(t)) P_m^{\mathrm{c},n}(t), \quad t \in [T], \tag{41}$$

where we define $P_m^{\mathrm{c},n}(t) \triangleq \|\tilde{\boldsymbol{g}}_m^n(\boldsymbol{\theta}_t)\|_2^2$. Given $\lambda^{\mathrm{c}}(t)$, we set, for $m \in [M], n \in [N], t \in [T]$,

$$\gamma_m^{\mathrm{c},n}(t) = \sigma \left(\frac{\bar{P}^n(t)}{\mathrm{E}_1(\lambda^{\mathrm{e}}(t)) P_m^{\mathrm{c},n}(t)}\right)^{1/2}, \tag{42}$$

and the PS computes

$$\bar{\gamma}^{\mathrm{c},n}(t) \triangleq \frac{1}{M} \sum_{m=1}^M \gamma_m^{\mathrm{c},n}(t), \quad n \in [N], t \in [T], \tag{43}$$

after receiving $\gamma_m^{\mathrm{c},n}(t)$.

By substituting $\boldsymbol{x}_m^n(\boldsymbol{\theta}_t)$ and $\boldsymbol{\alpha}^{\mathrm{c}}(t)$ into (4), it follows that, for $i \in [s]$, $n \in [N]$

$$\begin{aligned} y_i^n(t) &= \sum_{m \in \mathcal{M}_i^n(t)} \gamma_m^{\mathrm{c},n}(t) \big(\tilde{g}_{m,2(n-1)s+i}(\boldsymbol{\theta}_t) \\ &\quad + j\tilde{g}_{m,(2n-1)s+i}(\boldsymbol{\theta}_t)\big) + z_i^n(t) \\ &= \boldsymbol{a}_{2(n-1)s+i}^T \sum_{m \in \mathcal{M}_i^n(t)} \gamma_m^{\mathrm{c},n}(t) \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t) \\ &\quad + j\boldsymbol{a}_{(2n-1)s+i}^T \sum_{m \in \mathcal{M}_i^n(t)} \gamma_m^{\mathrm{c},n}(t) \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t) + z_i^n(t), \end{aligned} \tag{44}$$

where $\boldsymbol{a}_i^T$ denotes the i-th row of measurement matrix $\boldsymbol{A}$, and we note that $\tilde{g}_{m,i}(\boldsymbol{\theta}_t) = \boldsymbol{a}_i^T \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$, $i \in [\tilde{s}]$. The PS wants to recover $\frac{1}{M} \sum_{m=1}^M \boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$ from its noisy observations in (44). For this, using its knowledge of matrix $\boldsymbol{A}$ and the CSI, PS employs the approximate message passing (AMP) algorithm [34]. The AMP algorithm is represented by $\mathrm{AMP}_{\boldsymbol{A}}$ in Algorithm 1. The PS first obtains, for $i \in [s]$, $n \in [N]$,

$$\hat{y}_{2(n-1)s+i}(t) = \begin{cases} \frac{\mathrm{Re}\{y_i^n(t)\}}{\bar{\gamma}_m^{\mathrm{c},n}(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{45a}$$

$$\hat{y}_{(2n-1)s+i}(t) = \begin{cases} \frac{\mathrm{Im}\{y_i^n(t)\}}{\bar{\gamma}_m^{\mathrm{c},n}(t)|\mathcal{M}_i^n(t)|}, & \text{if } |\mathcal{M}_i^n(t)| \neq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{45b}$$

and then estimates

$$\hat{\boldsymbol{g}}^{\mathrm{c}}(\boldsymbol{\theta}_t) = \mathrm{AMP}_{\boldsymbol{A}}(\hat{\boldsymbol{y}}(t)), \tag{46}$$

where we define $\hat{\boldsymbol{y}}(t) \triangleq [\hat{y}_1(t), \cdots, \hat{y}_{\tilde{s}}(t)]^T$. If $\hat{\boldsymbol{y}}(t) \neq \mathbf{0}$, $\hat{\boldsymbol{g}}^{\mathrm{c}}(\boldsymbol{\theta}_t)$ is used to update the parameter vector as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{\boldsymbol{g}}^{\mathrm{c}}(\boldsymbol{\theta}_t)$. On the other hand, if $\hat{\boldsymbol{y}}(t) = \mathbf{0}$, the previous parameter vector is simply used as the new one, i.e., $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$.

Remark 4. *For $N = \lceil d/2s \rceil$, in which the entire gradient vectors are transmitted to the PS at each iteration, the CA-DSGD scheme reduces to the ECESA-DSGD scheme.*

Remark 5. *We remark that k is a design parameter which can take different values limited to $k < \tilde{s}$. For relatively*

small k values, $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$ can be recovered from $\frac{1}{M}\sum_{m=1}^{M}\tilde{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$ in a more reliable manner as we have more noisy measurements of the sparsified gradient estimate than the number of its non-zero elements; however, $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$ provides a less accurate estimate of the actual average gradient $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m(\boldsymbol{\theta}_t)$ as we have removed more componenets from the gradient estimates. On the other hand, for a relatively high k value, $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$ provides a better estimate for $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m(\boldsymbol{\theta}_t)$; however, it becomes harder to recover $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$ from $\frac{1}{M}\sum_{m=1}^{M}\tilde{\boldsymbol{g}}_m(\boldsymbol{\theta}_t)$ in a reliable manner.

Remark 6. *We note that, even though each device transmits a sparse vector $\boldsymbol{g}_m^{sp}(\boldsymbol{\theta}_t)$, their sum received over the channel does not need to be sparse. However, when the datasets are i.i.d. across devices and B is large, we expect the gradient estimates across devices to be statistically uniform, and thus, have similar sparsity patterns. Note, however, that the proposed CA-DSGD scheme does not require data to be independent across devices. As it will be shown in Fig. 2, the CA-DSGD scheme converges even when the local datasets are biased, in which case the sparsity patterns are expected to be more diverse. We observe that the transmissions from multiple devices still align on a small number of coordinates thanks to the superposition property of analog transmission. Thus, AMP still manages to recover the average gradient with reasonable accuracy, and the DSGD process converges, albeit more slowly compared to the IID scenario. We will see in Fig. 2 that CA-DSGD outperforms alternative analog and digital schemes with even a higher performance gap in the non-IID scenario.*

Remark 7. *With ESA-DSGD, each device transmits only the entries of its estimated gradient whose corresponding channel conditions are sufficiently good. Thus, the gradient vector is inherently sparsified, but only based on the channel gains, regardless of the importance of the gradient entries. Then the entire gradient vector is sent over the bandwidth-limited wireless MAC over orthogonal time periods. On the other hand, with CA-DSGD, each device sends only $k \leq \tilde{s}$ important gradient entries, where the magnitude of each entry is regarded as the importance metric, by projecting the sparse gradient vector to a low-dimensional vector of length $\tilde{s} \leq d$. We further highlight the error accumulation technique incorporated into ECESA-DSGD and CA-DSGD, whereas with ESA-DSGD, entries of the gradient vectors that are not sent are forgotten.*

Remark 8. *We highlight that, thanks to the wireless MAC providing a noisy version of the average of the gradient estimates, the analog schemes can potentially help preserve the privacy as well. This is particularly compelling for the proposed CA-DSGD scheme, where the gradient estimates are compressed through linear projection before transmission.*

V. NUMERICAL EXPERIMENTS

Here we compare the performances of the presented wireless edge learning schemes for the task of image classification. We run experiments on the MNIST dataset [35] with 60000 training and 10000 test samples, and train a single layer neural

network with $d = 7850$ parameters utilizing ADAM optimizer [36]. Throughout the experiments, we consider $\sigma^2 = 1$, and $s = \lceil d/20 \rceil$ parallel subchannels, which results in $N = 10$ for ESA-DSGD and ECESA-DSGD; and for any $\tilde{s}$ of the CA-DSGD scheme, we set the sparsity level to $k = \lfloor \tilde{s}/2.5 \rfloor$. For a fair comparison between the analog DSGD schemes, we set $\lambda^x(t) = \lambda, \forall x \in \{e,v,c\}$, and for average transmit power $\bar{P}^n(t)$ at the n-th time slot, $n \in [N]$, of the t-th iteration, $t \in [T]$, we calculate values of $\gamma_m^{e,n}(t)$, $\gamma_m^{v,n}(t)$ and $\gamma_m^{c,n}(t)$ for ESA-DSGD, ECESA-DSGD and CA-DSGD through (22), (32) and (42), respectively. Also, we consider $\bar{P}^n(t) = \bar{P}$, $\forall n,t$, for the analog schemes, and $\bar{P}_{m^*(t)}(t) = \bar{P}$, $\forall t$, for the digital schemes. The performance is measured as the accuracy with respect to the test data samples, called *test accuracy*, versus the normalized time Nt.

We consider two scenarios to model the data distribution across the devices: in *IID data distribution*, B randomly selected training data samples are assigned to each device at the beginning of training; while in *non-IID data distribution*, each device has B training data samples, where half of them are selected at random from only one class/label; that is, for each device, we first select two classes/labels at random, and then randomly select $B/2$ data samples from each of the two classes/labels. At each iteration, devices use all the B local data samples to compute their gradient estimates, i.e., the batch size is equal to the size of the local datasets.

For numerical comparison, we also consider the *error-free shared link* approach, where at each time slot the PS receives the average of the gradient estimates, $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{g}_m(\boldsymbol{\theta}_t)$, in a noiseless fashion. We note that, for the error-free shared link approach, we have $N = 1$. We consider three alternative digital schemes employing sparse binary compression [18], QSGD [8] and SignSGD [12] algorithms for gradient compression, respectively. When we refer to D-DSGD it refers to using the sparse binary compression technique. For a fair comparison, we apply QSGD and SignSGD to a limited number of gradient entries such that the final number of bits does not exceed the capacity of the underlying fading MAC. To be more precise, considering the device scheduling policy given in (6), $q^S(t)$ and $q^Q(t)$ gradient entries with highest magnitudes are selected for transmission for SignSGD and QSGD, respectively, while all the other entries are set to zero. With the SignSGD algorithm [12], the scheduled device transmits only the signs of the $q^S(t)$ selected entries, and a total of

$$r^S(t) = \log_2\binom{d}{q^S(t)} + q^S(t) \text{ bits, } \forall t, \quad (47)$$

are required to send the sign and location of each selected entry, and $q^S(t)$ is set as the largest integer satisfying $r^S(t) \leq R(t)$. With the QSGD algorithm [8], the scheduled device transmits a quantized version of each of the $q^Q(t)$ selected entries with a quantization level of 2^{l^Q}, the l_2-norm of the resultant vector with $q^Q(t)$ non-zero entries, and the locations of the non-zero entries. Thus, a total of

$$r^Q(t) = 32 + \log_2\binom{d}{q^Q(t)} + (1 + l^Q)q^Q(t) \text{ bits, } \forall t, \quad (48)$$

(a) IID data distribution

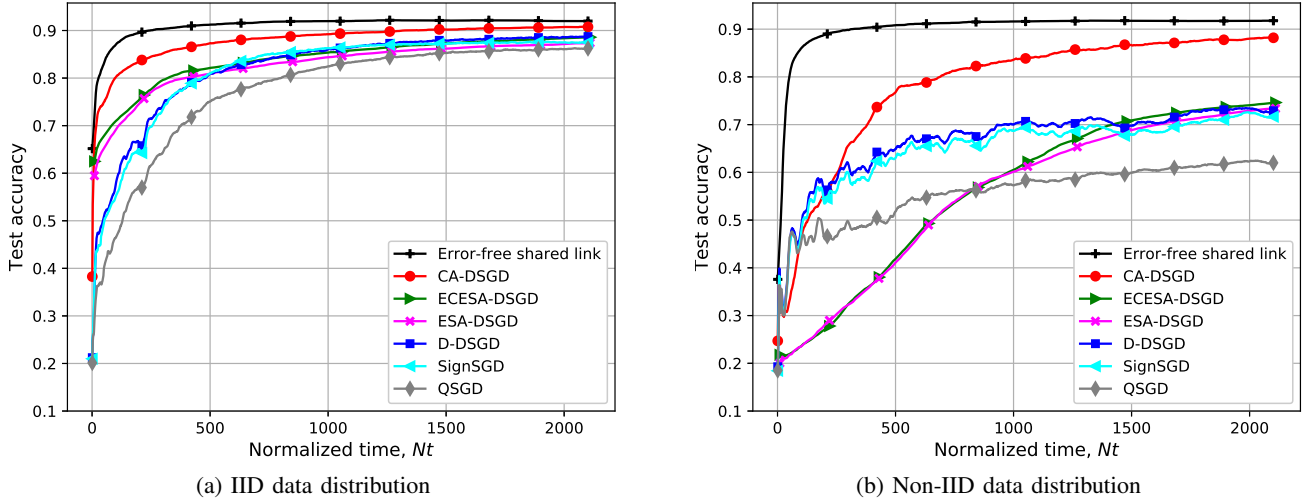(b) Non-IID data distribution

Fig. 2: Test accuracy for IID and non-IID data distribution scenarios for $M = 25$, $B = 1000$ and $\bar{P} = 20$.

are sent over the wireless fading MAC, where $q^{\mathrm{Q}}(t)$ is set as the largest integer satisfying $r^{\mathrm{Q}}(t) \leq R_t$. Here we consider a quantization level $l^{\mathrm{Q}} = 2$ for QSGD.

We further consider the OD-DSGD scheme, where each device has access to $\lfloor s/M \rfloor$ distinct subchannels to perform digital transmission without interfering with other devices. Due to symmetry across devices, we allocate subchannels $(m-1)\lfloor s/M \rfloor + 1$ to $m \lfloor s/M \rfloor$ to device m, $m \in [M]$. Similarly to the D-DSGD scheme, we use the capacity upperbound to determine the number of bits each user can convey to the PS at each iteration. This bound is computed by waterfilling across the $\lfloor s/M \rfloor$ channels available to each device as in (9)-(11). We use sparse binary compression with the OD-DSGD scheme as well with the sparsity level $q_m^{\mathrm{O}}(t)$ set as the largest integer satisfying $\log_2 \binom{d}{q_m^{\mathrm{O}}(t)} + 33 \leq R_m^{\mathrm{O}}(t)$, $\forall t$.

In Fig. 2, we compare the performances of different analog and digital schemes for IID and non-IID data distribution scenarios, for $M = 25$ devices, $B = 1000$ training data samples and average transmit power constraint $\bar{P} = 20$. We consider $\tilde{s} = 2s = d/10$, i.e., $N = 1$ for CA-DSGD, and we set the threshold value to $\lambda = 10^{-3}$. Observe that for both IID and non-IID data distribution cases CA-DSGD outperforms all other analog and digital schemes with the improvement substantially larger for non-IID data distribution, which shows its robustness to bias in the data distribution. CA-DSGD has a smaller convergence speed in non-IID case which is due to the reduction in the similarity of the sparsity patterns of the gradients across devices, although it does converge much faster and to a much higher accuracy level compared to the other schemes under consideration. The gap between the error-free shared link approach and CA-DSGD is relatively small for the IID case, and the final test accuracies of the two approaches are also similar for the non-IID case, although CA-DSGD converges more slowly in this case. Unlike the digital schemes, CA-DSGD benefits from the superposition property of the underlying wireless MAC by aligning the transmit powers to dominate the noise. We further highlight that the main reasons for the degradation of the ESA-DSGD over CA-DSGD are i) scheduling gradient entries for transmission

only based on the channel gains; ii) transmitting the entire gradient vectors of relatively huge dimensions (compared to the channel bandwidth); iii) ignoring the gradient entries which have not been transmitted due to the poor conditions of their corresponding channels. We note that ECESA-DSGD resolves the last issue by utilizing error accumulation technique, which provides some gains with respect to ESA-DSGD, but we observe that better scheduling and more efficient utilization of the bandwidth through linear projection provide significant gains, especially for the non-IID case, where the performances of ESA-DSGD and ECESA-DSGD significantly degrade in terms of the test accuracy, as well as the convergence speed. Also, D-DSGD provides a better accuracy than SignSGD and QSGD for both data distribution scenarios, and the performance of all the digital schemes under consideration deteriorate substantially in the non-IID case; this performance loss is more severe for QSGD.

In the following experiments, we only consider IID data distribution. In Fig. 3, we compare the performances of different analog and digital algorithms for two different average transmit power values $\bar{P} = 10$ and $\bar{P} = 30$. We consider $M = 25$ and $B = 10$, and we set $\tilde{s} = 2s = d/10$, i.e., $N = 1$ for CA-DSGD, and $\lambda = 5 \times 10^{-3}$. As it can be seen, CA-DSGD continues to outperform all the other schemes, with a relatively small gap to the error-free shared link approach. By comparing Figures 3a and 3b, it can be seen that the performances of all the schemes improve with $\bar{P}$, but the improvement is more significant for the digital schemes in terms of both the accuracy and the convergence speed, except QSGD which only improves in terms of accuracy. This shows that the analog schemes are less sensitive to a reduction in the average transmit power than the digital ones. This is because, thanks to the signal-superposition property, the system continues to operate in a relatively high effective signal-to-noise ratio (SNR) regime despite reduction in the transmission power of individual devices.

In Fig. 4, we compare the performances of analog and digital schemes for different (M, B) pairs, $(M, B) \in \{(15, 2000), (30, 1000)\}$, both having the same size of training
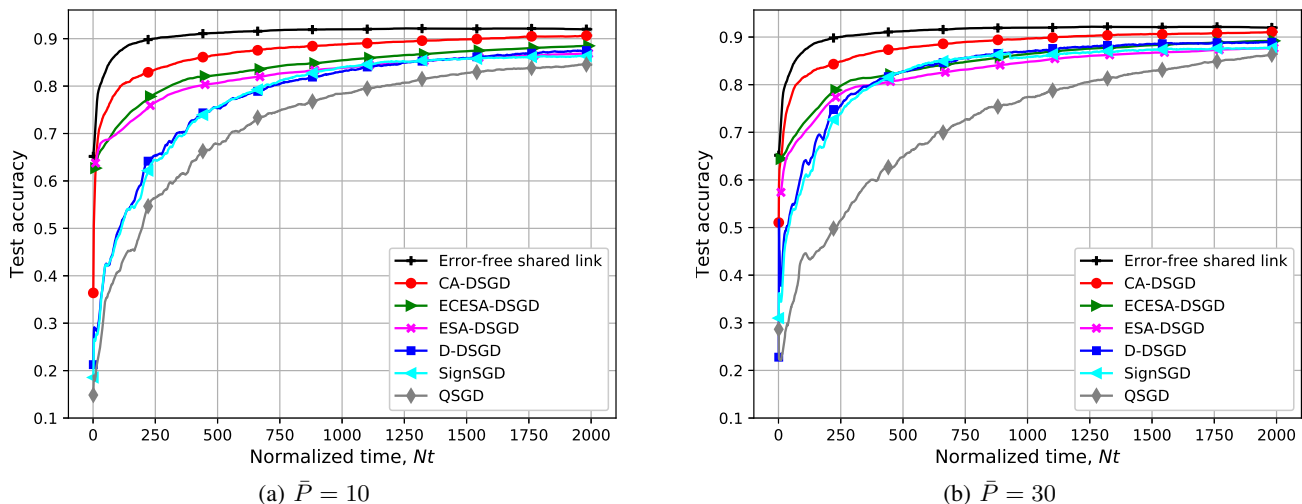
(a) $\bar{P} = 10$

(b) $\bar{P} = 30$

Fig. 3: Test accuracy of different algorithms for two different $\bar{P}$ values, $\bar{P} \in \{10, 30\}$, when $M = 25$ and $B = 1000$.

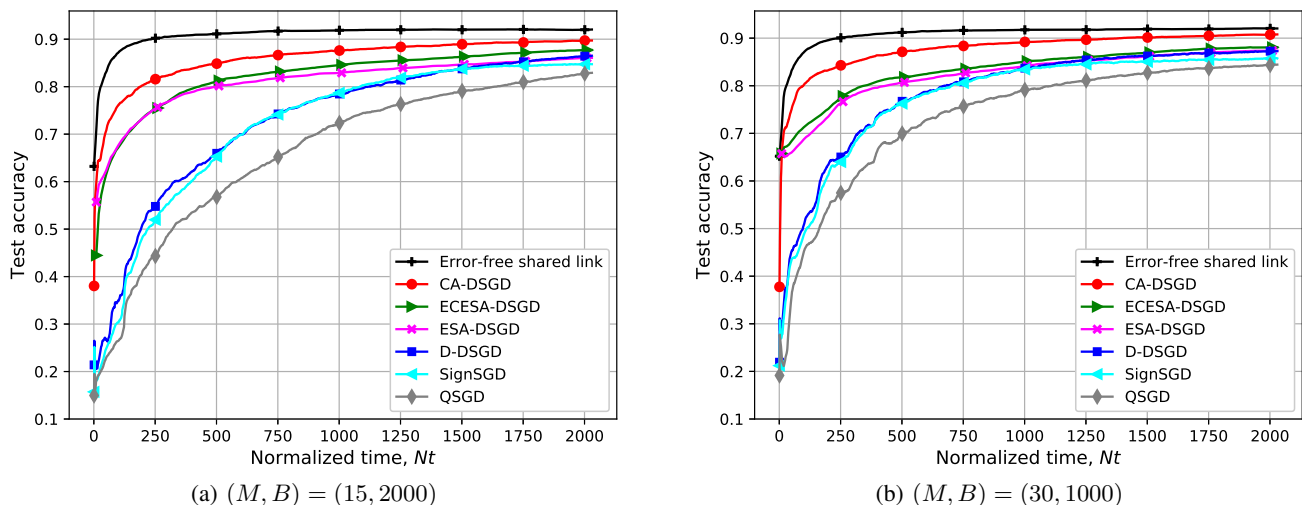
(a) $(M, B) = (15, 2000)$

(b) $(M, B) = (30, 1000)$

Fig. 4: Test accuracy of different algorithms different (M, B) pairs when MB is fixed, and $\bar{P} = 10$.

data in total. We consider $\bar{P} = 10$, and we set $\lambda = 5 \times 10^{-3}$, and $\tilde{s} = 2s = d/10$, which results in $N = 1$ for CA-DSGD. It is again evident that CA-DSGD outperforms all the other schemes with the improvement over ESA-DSGD and ECESA-DSGD more noticeable for the higher M value. As it can be seen, the performances of the analog schemes improve with M, since increasing M provides additional power introduced by each device and increases the robustness of the estimation against noise. We note that this improvement is larger for CA-DSGD, which is due to the more efficient utilization of the gradient estimates computed by the devices. Digital schemes also gain from increasing M, which is due to the additional power allocated to the selected device, as the devices are less frequently scheduled for transmission. We note that the superiority of the ECESA-DSGD over ESA-DSGD reduces with M, which shows that error accumulation is less effective for higher M values when MB is fixed. This is because for larger M, the chance of receiving more estimates for each entry of the actual gradient vector is higher (each gradient entry is estimated more accurately at the PS), and it is less

likely that no estimate of any gradient entry is received by the PS. Accordingly, the benefit of error accumulation becomes less significant for higher number of devices.

In Fig. 5, we compare the performance of D-DSGD with that of OD-DSGD for different $\bar{P}$ values, $\bar{P} \in \{20, 100\}$, when $M = 25$ and $B = 1000$. For both power values we observe that D-DSGD significantly outperforms OD-DSGD in terms of accuracy and convergence speed, while the superiority is more highlighted for the higher $\bar{P}$ value. This shows that opportunistically allocating all the available bandwidth to only a single device is better than sharing it equally among all the devices which indicates that it is better to receive an accurate gradient estimate from a single device at each iteration, instead of receiving coarse estimates from all the devices. This improvement is more significant when $\bar{P}$ increases.

In Fig. 6 we investigate the impact of $\tilde{s}$ on the performance of CA-DSGD. We consider $\tilde{s} \in \{2s, 4s\} = \{d/10, d/5\}$ for CA-DSGD, in which $\tilde{s} = 2s$ and $\tilde{s} = 4s$ are equivalent to $N = 1$ and $N = 2$, respectively. We have $M = 15$, $B = 1000$ and $\bar{P} = 1$, ans we set $\lambda = 5 \times 10^{-3}$.
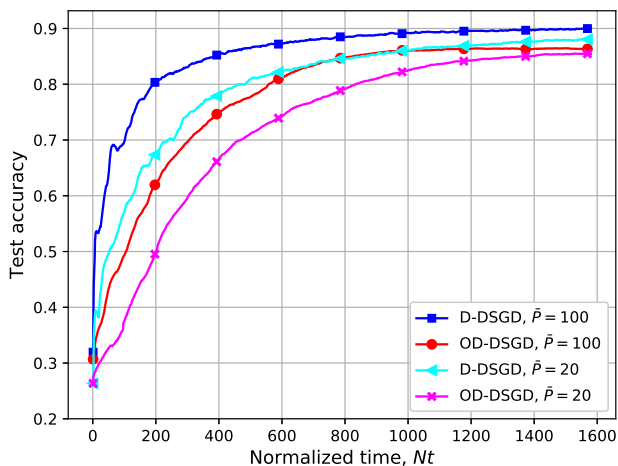
Fig. 5: Test accuracies of D-DSGD and OD-DSGD schemes for different $\bar{P}$ values, $\bar{P} \in \{20, 100\}$, when $M = 25$ and $B = 1000$.
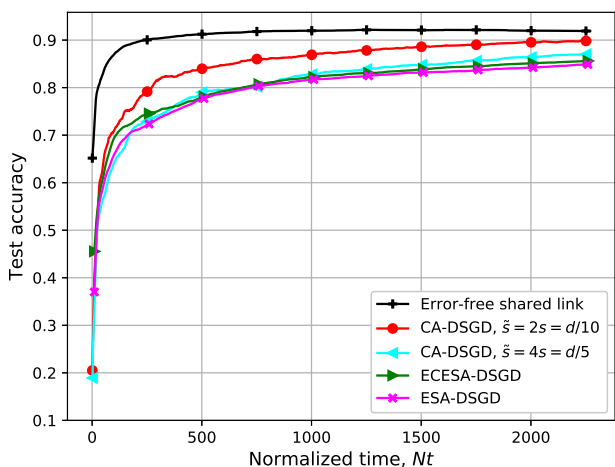

Fig. 6: Test accuracies of different analog schemes with different $\tilde{s}$ values, $\tilde{s} \in \{2s, 4s\}$, for the CA-DSGD scheme, when $M = 25$, $B = 1000$, $\bar{P} = 1$.
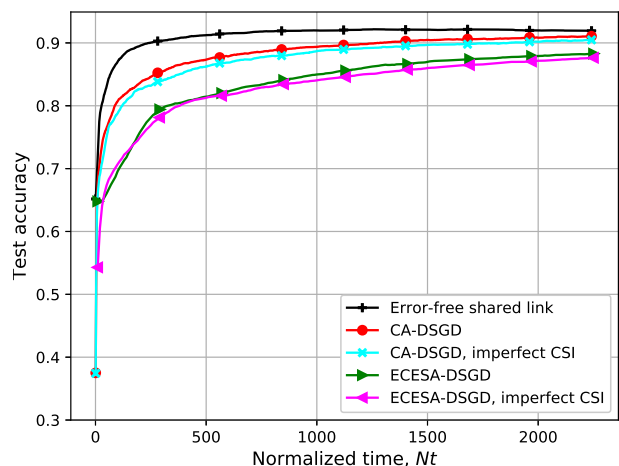

Fig. 7: Test accuracies of CA-DSGD and ECESA-DSGD schemes under imperfect CSI $\hat{h}_{m,i}^n(t) = h_{m,i}^n(t) + \tilde{h}_{m,i}^n(t)$, $\forall m, n, i, t$, with $\tilde{h}_{m,i}^n(t)$ i.i.d. according to $\mathcal{CN}(0, 1)$, when $M = 25$, $B = 1000$ and $\bar{P} = 10$.

We highlight the superiority of the CA-DSGD scheme for both $\tilde{s}$ values under consideration over ESA-DSGD. The ECESA-DSGD scheme, which is equivalent to CA-DSGD for $\tilde{s} = d$, also outperforms ESA-DSGD slightly. However, as it can be seen, the performance of CA-DSGD degrades as $\tilde{s}$ increases, which indicates that transmitting more sparse versions of the gradient estimates while using the available channel bandwidth for further iterations results in a higher accuracy. The flexibility in choosing the dimension of the transmitted gradient estimates makes the proposed CA-DSGD scheme particularly compelling for learning at the wireless edge under strict bandwidth limit.

In Fig. 7, we consider the impact of imperfect CSI on the performance of analog schemes CA-DSGD and ECESA-DSGD for $M = 25$, $B = 1000$ and $\bar{P} = 10$. We set $\lambda = 5 \times 10^{-3}$, and $\tilde{s} = 2s = d/10$, which results in $N = 1$ for CA-DSGD. We assume a noisy CSI at device m given by $\hat{h}_{m,i}^n(t) = h_{m,i}^n(t) + \tilde{h}_{m,i}^n(t)$, $\forall m, n, i, t$, where $\tilde{h}_{m,i}^n(t)$ is i.i.d. according to $\mathcal{CN}(0, 1)$, i.e., a complex normal random

variable with the same variance as the actual channel gain $h_{m,i}^n(t)$. We note that all the processing at the devices, such as power allocation, finding the set $\mathcal{M}_i^n(t)$, and obtaining $\gamma_m^{v,n}(t)$ and $\gamma_m^{c,n}(t)$, and consequently $\bar{\gamma}^{v,n}(t)$ and $\bar{\gamma}^{c,n}(t)$ for CA-DSGD and ECESA-DSGD, respectively, are performed based on the imperfect CSI $\hat{h}_{m,i}^n(t)$, $\forall m, n, t$. As it can be seen, both the CA-DSGD and ECESA-DSGD are robust against the imperfect CSI. To be more precise, after 2250 time slots (2250 SGD iterations for CA-DSGD and 225 SGD iterations for ECESA-DSGD), the final test accuracy reduction for CA-DSGD due to the imperfect CSI is 0.67%, and that of ECESA-DSGD is 0.76%. We highlight that, with imperfect CSI, even though the users will allocate higher or lower power to each subchannel than the optimal one, the cumulative effect becomes negligible since these variations across users are averaged out thanks to the superposition property.

VI. CONCLUSIONS

We have studied FL at the wireless edge, where M devices with limited transmit power and datasets communicate with the PS over a bandwidth-limited fading MAC to minimize a loss function by performing DSGD. The PS updates the parameter vector, and shares it with the devices over an error-free shared link. We first presented a digital approach that treats computation and communication separately. At each iteration of the proposed D-DSGD scheme, one device is selected depending on the channel states, and the selected device first quantizes its gradient estimate, and transmits the quantized bits to the PS using a capacity-achieving channel code. Then we studied an alternative analog transmission approach, which does not employ quantization or channel coding, and exploits the superposition property of the wireless MAC, rather than orthogonalizing the transmissions from different devices. We have proposed the CA-DSGD scheme, where each device employs gradient sparsification with error accumulation followed by linear projection to reduce the typically very large parameter vector dimension to the limited channel bandwidth.

We have also designed a power allocation scheme to align the received vectors at the PS while satisfying the average power constraints at the devices. The CA-DSGD scheme allows a much more efficient use of the limited channel bandwidth, and benefits from the "beamforming effect" thanks to the similarity in the patterns of the gradient estimates across devices. The impact of various system parameters on the performance is studied numerically considering MNIST classification across edge devices as an example. Numerical results show that CA-DSGD outperforms D-DSGD and other state-of-the-art analog schemes consistently, while this improvement is even more significant for the non-IID data distribution scenario.

REFERENCES

[1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv:1610.05492v2 [cs.LG]*, Oct. 2017.

[2] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *[online]. Available. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html*, Apr. 2017.

[3] J. Konecny and P. Richtarik, "Randomized distributed mean estimation: Accuracy vs communication," *arXiv:1611.07555 [cs.DC]*, Nov. 2016.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

[6] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *ICML*, Jul. 2015.

[7] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *INTERSPEECH*, Singapore, Sep. 2014, pp. 1058–1062.

[8] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via randomized quantization and encoding," in *NIPS*, Long Beach, CA, Dec. 2017, pp. 1709–1720.

[9] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "TernGrad: Ternary gradients to reduce communication in distributed deep learning," *arXiv:1705.07878v6 [cs.LG]*, Dec. 2017.

[10] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv:1606.06160v3 [cs.NE]*, Feb. 2018.

[11] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "ATOMO: Communication-efficient learning via atomic sparsification," *arXiv:1806.04090v2 [stat.ML]*, Jun. 2018.

[12] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," *arXiv:1802.04434v3 [cs.LG]*, Aug. 2018.

[13] B. Li, W. Wen, J. Mao, S. Li, Y. Chen, and H. Li, "Running sparse and low-precision neural network: When algorithm meets hardware," in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jeju, South Korea, Jan. 2018.

[14] N. Strom, "Scalable distributed DNN training using commodity gpu cloud computing," in *INTERSPEECH*, 2015.

[15] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv:1704.05021v2 [cs.CL]*, Jul. 2017.

[16] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv:1712.01887v2 [cs.CV]*, Feb. 2018.

[17] X. Sun, X. Ren, S. Ma, and H. Wang, "meProp: Sparsified back propagation for accelerated deep learning with reduced overfitting," *arXiv:1706.06197v4 [cs.LG]*, Oct. 2017.

[18] F. Sattler, S. Wiedemann, K. Muller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," *arXiv:1805.08768v1 [cs.LG]*, May 2018.

[19] C. Renggli, D. Alistarh, T. Hoefler, and M. Aghagolzadeh, "SparCML: High-performance sparse communication for machine learning," *arXiv:1802.08021v2 [cs.DC]*, Oct. 2018.

[20] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli, "The convergence of sparsified gradient methods," *arXiv:1809.10505v1 [cs.LG]*, Sep. 2018.

[21] Y. Tsuzuku, H. Imachi, and T. Akiba, "Variance-based gradient compression for efficient distributed deep learning," *arXiv:1802.06058v2 [cs.LG]*, Feb. 2018.

[22] S. U. Stich, "Local SGD converges fast and communicates little," *arXiv:1805.09767v2 [math.OC]*, Jun. 2018.

[23] T. Lin, S. U. Stich, and M. Jaggi, "Don't use large mini-batches, use local SGD," *arXiv:1808.07217v3 [cs.LG]*, Oct. 2018.

[24] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," *arXiv:1805.09965 [stat.ML]*, May 2018.

[25] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, to appear, Feb. 2020.

[26] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," *arXiv:1812.11494 [cs.IT]*, Jan. 2019.

[27] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *arXiv:1812.11750 [cs.LG]*, Jan. 2019.

[28] M. M. Amiri and D. Gündüz, "Over-the-air machine learning at the wireless edge," in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, Jul. 2019, pp. 1–5.

[29] S. Jakubczak and D. Katabi, "SoftCast: One-size-fits-all wireless video," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 449–450, Oct. 2010.

[30] R. Xiong, J. Zhang, F. Wu, J. Xu, and W. Gao, "Power distortion optimization for uncoded linear transformed transmission of images and videos," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 222–236, Jan. 2017.

[31] T-Y. Tung and D. Gündüz, "SparseCast: Hybrid digital-analog wireless image transmission exploiting frequency-domain sparsity," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2451–2454, Dec. 2018.

[32] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge, UK: Cambridge University Press, 2005.

[33] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[34] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[35] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/*, 1998.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980v9 [cs.LG]*, Jan. 2017.

Mohammad Mohammadi Amiri (S'16) received the B.Sc. and M.Sc. degrees in Electrical Engineering from Iran University of Science and Technology in 2011 and University of Tehran in 2014, respectively, both with highest rank in classes. He also obtained a Ph.D. degree at Imperial College London in 2019. He is currently a Postdoctoral Research Associate in the Department of Electrical Engineering at Princeton University. His research interests include information and coding theory, machine learning, wireless communications, and signal processing.

Deniz Gündüz (S'03-M'08-SM'13) received the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively. After his PhD, he served as a postdoctoral research associate at Princeton University, and as a consulting assistant professor at Stanford University. He was a research associate at CTTC in Barcelona, Spain until September 2012, when he joined the Electrical and Electronic Engineering Department of Imperial College London, UK, where he is currently a Reader (Associate Professor), and leads the Information Processing and Communications Laboratory (IPC-Lab). His research interests lie in the areas of communications, information theory, machine learning, and privacy.